

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2001-256064

(43) Date of publication of application : 21.09.2001

(51)Int.Cl. G06F 9/46
G06F 15/18

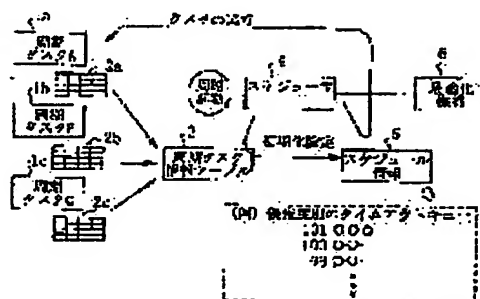
(72)Inventor : KAWAKAMI TAKESHI

(54) OPTIMIZATION SCHEDULING SYSTEM OF PLURAL PERIODIC PERFORMANCE TASKS

(57)Abstract

PROBLEM TO BE SOLVED: To provide a scheduling system to enable optimization so that width of shift of intervals of performance periods of plural periodic tasks to perform a constant periodic processing is minimized.

SOLUTION: The optimization of scheduling to enable performance of the plural periodic tasks to perform the constant periodic processing is performed at constant periodic intervals by optimizing a performance schedule to perform the plural periodic tasks 1 by optimization mechanism 6 based on intrinsic information by every one of the plural periodic tasks 1 to perform the constant periodic processing and correcting the performance schedule of the plural preset periodic tasks 1 according to the optimized performance schedule by a scheduler 4.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

THIS PAGE BLANK (USPTO)

(A) 公報 (12) 公開

特開2001-256064
(P2001-256064A)

(43)公開日 平成13年9月21日(2001.9.21)

識別記号
340
550

F 1	7-73-1* (8-25)
G 06 F	3 40 E 5 B 0 9 8
15/18	5 50 C

審査請求 未請求 請求項の数11 OL (全22頁)

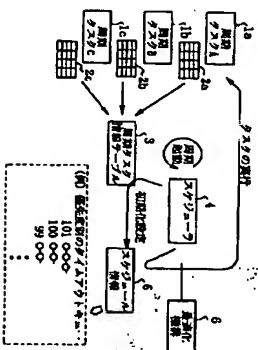
(21) 出願番号	特願2000-664051 P2000-664051
(22) 出願日	平成12年3月10日 (2000.3.10)
(71) 出願人	000008013 三菱電機株式会社 東京都千代田区丸の内二丁目2番3号
(72) 発明者	川上 茂 東京都千代田区丸の内二丁目2番3号 三
(74) 代理人	100102839 三菱電機株式会社内 弁理士 宮田 金雄 (外1名) Fターム(参考) 5B08 FT04 G404 G408 G505

(54) 【発明の名称】 複合周期実行タスクの最適化スケジューリング方式

(57) 【要約】

【課題】 定周期処理を実行する複数の周期タスクの実行周期の間隔が最小限となるような最適化が可能なスケジューリング方式を提供する。

【解決手段】 定期利息連と実行する複数の期間タスク1毎の留付情報に基づいて、最速化候補6で複数の期間タスク1を実行する実行スケジュールの最速化を行い、スケジュール4でこの最速化された実行スケジュールに従って、予め決定された複数の期間タスク1の実行スケジュールを修正することにより、定期利息連と実行する複数の期間タスクを一定の期間内で実行できるようにスケジュールングの最速化を行うことができる。



【特許請求の範囲】

【請求項1】 定周期処理を実行する複数の周期タスク毎の固有情報を格納するテーブルと、

上記テーブルの上記項目情報に基づいて、上記複数の局
 群データが実行する定期処理の一回あたりの周期間隔を
 一定にする実行スケジュールの最適化が必要か否かの判
 定を行い、最適化が必要ときには実行スケジュールの
 最適化を行う最適化手段と、

上記最適化手段により最適化された実行スケジュームに
基づいて、予め設定された上記演算の間隔タスクの実行
スケジュームを修正するスケジューラと、備えたこと
を特徴とする複数同期実行タスクの最適化スケジューリ
ング方式。

【請求項2】 上記図形情報は、上記第2数の周相タスク毎に予め設定されている上記第2数の周相タスクを実行する第1の周相時間情報と、上記第2数の周相タスクを実行するときに生じる上記第1の周相時間情報の変化量を示す周相時間情報と、上記第2数の周相タスクが前回実行されたときの第2の周相時間情報とを備え、

上記最速手続は、上記問題時刻情報の値が0から0以外
かを判断し、上記問題時刻情報が0以外の時には上記第
2の所算時刻判定と上記第1の所算時刻情報とが一致し
ていない否かを判定し、一致していないときには、上記
問題時刻情報に基づいて上記第1の所算時刻情報の変更
を行うことを特徴とする請求項1記載の所算時刻算行方
式の最適化システムコンピュータ方式。

【請求項3】 上記固有情報は、複数の問題時間情報を備え、

上記最適化手段は、上記模数の調整時間情報がいずれかに基づいて上記第1の周期時間情報の変更を行うことを特徴とする請求項2記載の積放同期実行タスクの最適化スケジューリング方式。

【請求項1】 上記遅延情報は、上記遅延数の当該スラッシュが前々回更新されたときの第3の同期時刻情報と前記第3の同期時刻情報と上記第2の同期時刻情報と上記第3の同期時刻情報との比較を行い、この比較結果に基づいて上記遅延部情報情報を設定し、この設定された遅延部時間情報に基づいて、上記第1の同期時刻情報の変更を行うことを特徴とする請求項2記載の装置、同期実行手段の最遅にスラッシュをリソソ方式。

[図表5] 上記第1の場所時間情報と上記第2の場所時間情報を与える第1のニューロン群と、この受入パターンジェネレーション時に必要となる第2の場所時間情報と出力される第2のニューロン群と、上記第1のニューロン群で入力された上記第1の場所時間情報と上記第2の場所時間情報とに基づいて第3のニューロン群に出力される第3の場所時間情報とを決定する第4の場所情報とを出力する第5のニューロン群と、第7のニューロン群で入力された上記第1の場所時間情報と上記第2の場所情報とを基にした第6のニューロン群で出力された情報の一部ともなう第8のニュー

ニューロン群と、第4のニューロン群から第3のニューロン群に出力される上記第2の問題時間情報を決定するための情報を導き出す第4のニューロン群と、から構成されるニューラルネットワークを備え、

上記最適化手段は、上記ニューラルネットワークにより出力された今回の実行スケジュール数値時に必要となる第2の問題時間情報に基づいて上記第1の問題時間情報の変更を行うことを特徴とする請求項2記載の複数問題解決手段の最適化スケジュールソング方式。

【請求項6】 上記複数の周期タスクの実行状況を確
 する利用者により命令が入力される命令入力手段を備
 え、

上記記載に準拠は、上記第1の期間時間情報の変更を行うか否かを示すフラグを格納した第1のメモリを備え、上記第2の発行要求からの命令に基づいて上記フラグが変更されたら、上記変更されたフラグに基づいて、上記第2の期間、スロットの変更スケジュールの最適化を行うことを特徴とする請求項1記載の装置、図10の要項に示すスケジューリング方式。

【請求項7】 上記最速に手段は、上記複数の周期タスク毎に周期タスクの最速にが全て終了する最速に実行時間情報を格納した第2のメモリを備え、この最速に実行時間情報に基づいて上記複数の周期タスクの実行スケジュールの最速にを行うことを特徴とする請求項1記載の、複数周期実行タスクの最速にスケジューリング方式。

【請求項8】 上記複数の周知タスクの実行スケジュールを、上記複数の周知タスクの実行状況を確認する利用者からの指示に基づいて最適化するアプリケーションソフトウェアが備えられたことを特徴とする請求項1記載の最適化手段を備えたことを特徴とする請求項1記載の最適化実行タスクの最適化スケジュール方式。

【請求項9】 上記アブリケーション側最適化手段は、上記データと、上記最適化手段と、上記スケジューラ

と、上座アツツグを結納する第1のメモリと、上座段葛に実行時間切れを付けた第2のメモリと同一機能を持つ実行時間切れを付けた第3のメモリとを、上座アツツグに渡す。それと、第3のメモリと、第3の段葛に渡す。第2の段葛アツツグと、上座アツツグを結納する第2のメモリと、第3の段葛に実行時間切れを付けた第3の段葛アツツグと、第3のメモリとを備えたことを特徴とする請求項8記載のプログラム。

【請求項10】 上座段葛に手段は、上座アツツグアツツグの動きを一時止する第4のメモリを起動し手段と、上座アツツグアツツグ起動し手段によるメモリの一時停止中に、上座アツツグアツツグの動きを手段で強制停止させた上座アツツグアツツグを、上座アツツグアツツグ決定された実行アツツグと変換する請求項9記載のプログラム内部要素と手段とを備えたことを特徴とする請求項9記載のプログラムを実行タスクの動きはアツツグアツツグ手段。

【請求項11】 上記アブリケーション最適化手段は、上記複数の周期タスクに新たな周期タスクが追加されたときに、この新規周期タスクの固有情報を格納し、

この新規同期タスクの固有情報に基づいて実行スケジュールの最適化を行い、この最適化した実行スケジュールを上記最適化手段に送ることを特徴とする請求項8記載の新規同期タスクの最適化スケジュール方式【発明の詳細な説明】

【0001】

【発明の属する技術分野】この発明は、定期処理を必要とする同期実行タスクを複数実行しているリアルタイムシステムのスケジューラに対して、各同期タスクの実行間の間隔が最小値となるような最適化を可能とするスケジュール方式に関するものである。

【0002】

【従来の技術】従来のスケジュール方式では、定期実行を必要とする同期タスクは、優先度及び実行頻度を基に、スケジュール化されている。例えば、特許平9-319597号公報に記載の1周期プロセッサのスケジューリング方法では、周期的に実行されるプロセス群を含むプロセスグループが複数個実行して実行される場合において、各プロセスグループのグループヘッダプロセッサ（グループ内で処理が先行であるプロセス）が、定期処理周期と一周毎の必要CPU時間を指定してCPU時間割当てを要求すると、他のプロセスグループのCPU割当て時間と競合しないように確保される。

【0003】プロセスグループの一つを起動する時点で、このグループ内のグループヘッダプロセッサの優先度が最高に変更され、実行される。このグループヘッダプロセッサは、他のプロセスを実行時に優先度を継承することにより実行し、つまり順次に優先度が降られ、それでは実行される。最後まで実行し終えると優先度を下げ、グループの実行権を他のグループに譲り渡す。

【0004】即ち、グループ内では、優先度によってのみ一週に実行の順番が確定してしまふ。前記な例で説明すると、全て実行度が同一の周期が10（単位はシステム最小タイムスロットとする）の同期タスクA、B、Dがあった場合、従来のスケジューリング方式では、周期時間の最短期間タスクからの割り当てが行われ、最長の同期タスクDで、同期間隔が一定にならない、いわゆるばら（ゆらぎ）が生じてしまふ。

【0005】例えば同期タスクAを基準にして、同期タスクAの1単位後に同期タスクBが開始され、次に同期タスクAの2単位後に同期タスクCが開始され、次に同期タスクAの3単位後に同期タスクDが開始される場合において、再度同期タスクAから同期タスクDまでの間隔タスクの差が行われる時に、同期タスクDで開始のタイムズリフが合わない、即ち同期間隔にばらが生じてしまふ。

【0006】

【発明が解決しようとする課題】以上説明したように従

来の周期的プロセスのスケジューリング方式では、以下に示すような課題があった。即ち、従来のスケジューリング方式では、必要とされている定期タスクをスケジュールする際、割り当てを開始するスタート地点がすべて一律であるため、同期タスクが3つあれば、システムの起動開始時には、3つ全ての同期タスクがキュー待ち状態となつてスケジュールが行われる。

【0007】また、優先度を減らすことにより、一部の定期タスクの間隔を短縮し一定にすることは可能であるが、この見返りとして、少しでも最高の優先度より低いものは保証されなくなってしまう。そのため、大抵の場合、定期処理を断ずるにせよ、間隔タスクは複数あり、CPUの使用状況から判断して、明らかに可能な場合であっても、このような従来のスケジューリング方式では阻害されてしまう可能性がある。

【0008】さらに、現存するものの同期タスクが同期タスクではなく、緊急に実行する必要がある突然発生された緊急同期タスクならば、定期タスクの妨けは一時的な間隔のぶれだけであるが、他の定期タスクが自分のより高い優先度をもつと、間隔のぶれは恒久的なものとなる。

【0009】この発明は上記のような諸点を解決するためになされたもので、一定の周期で実行しなければならない同期タスクを複数実行して実行する際、と同期タスクも一定の周期で定期処理が実行できるようにスケジュールの最適化を行うことを目的とする。

【0010】

【課題を解決するための手段】第1の発明は、定期処理を実行する複数の同期タスク毎の固有情報を格納するテーブルと、上記テーブルの上位固有情報に基づいて、上記複数の同期タスクが実行する定期処理の一周毎の同期間隔を一定にする実行スケジュールの最適化が必要否かの判定を行い、最適化が必要ときには実行スケジュールの最適化を行う最適化手段と、上記最適化手段により最適化された実行スケジュールに基づいて、予め設定された上記複数の同期タスクの実行スケジュールを修正するスケジューラと、を備えたものである。

【0011】第2の発明は、上記複数の同期タスク毎に予め設定されている上記複数の同期タスクを実行する第1の同期間隔情報と、上記複数の同期タスクを実行するとき生じる上記第1の同期間隔情報の変化量を調整する調整間隔情報と、上記複数の同期タスクが前回実行されたときの第2の同期間隔情報とを有する固有情報と、これらの同期間隔情報の値が0か0より大きいかを判断し、上記調整間隔情報が0以下の時には上記第2の同期間隔情報と上記第1の同期間隔情報とが一致しているかを判定し、一致していないときには、上記調整間隔情報に基づいて上記第1の同期間隔情報の変更を行う最適化手段とを備えたものである。

【0012】第3の発明は、複数の調整間隔情報を有す

る固有情報と、上記複数の調整間隔情報のいずれかに基づいて上記第1の同期間隔情報の変更を行う最適化手段とを備えたものである。

【0013】第4の発明は、上記複数の同期タスクが前回実行されたときの第3の同期間隔情報を有する固有情報と、上記第3の同期間隔情報と上記第3の同期間隔情報との比較を行い、この比較結果に基づいて上記調整間隔情報を設定し、この設定された調整間隔情報に基づいて上記第1の同期間隔情報の変更を行う最適化手段とを備えたものである。

【0014】第5の発明は、上記第1の同期間隔情報と上記調整間隔情報を入力する第1のニーロフ群と、次の同期実行スケジュール設定時に必要となる第2の調整間隔情報出力される第2のニーロフ群と、上記第1のニーロフ群で入力された上記第1の同期間隔情報と上記調整間隔情報とに基づいて第2のニーロフ群で出力される第3の調整間隔情報とを決定するための情報出力する第3のニーロフ群と、第1のニーロフ群で入力された上記第1の同期間隔情報と上記調整間隔情報と、上記第3のニーロフ群で出力された情報の一部とを入力する第4のニーロフ群と、第4のニーロフ群から

第3のニーロフ群に出力される上記第2の調整間隔情報を決定するための情報を導き出す第4のニーロフ群と、上記第4のニーロフ群で出力される上記第2の調整間隔情報と、上記第3のニーロフ群で入力された上記第1の同期間隔情報と上記調整間隔情報と、上記第3のニーロフ群で出力された情報の一部とを入力する第4のニーロフ群と、第4のニーロフ群から

【0015】第6の発明は、上記複数の同期タスクの実行状況を監視する利用者により命令が与えられる命令入力手段を備え、上記最適化手段は、上記第1の同期間隔情報の変更を行うか否かを示すフラグを格納した第1のメモリを備え、上記命令実行手段からの命令に基づいて上記フラグが変更され、この変更されたフラグに基づいて、上記複数の同期タスクの実行スケジュールの最適化を行うものである。

【0016】第7の発明は、上記複数の同期タスク毎に同期タスクの最適化に全て共通する最適化実行間隔情報を格納した第2のメモリを備え、この最適化実行間隔情報に基づいて上記複数の同期タスクの実行スケジュールの最適化を行う最適化手段を備えたものである。

【0017】第8の発明は、上記複数の同期タスクの実行スケジュールを、上記複数の同期タスクの実行状況を監視する利用者からの指示に基づいて最適化するアナログスケジューラを備えたものである。

【0018】第9の発明は、上記テーブルと、上記最適化手段と、上記スケジューラと、上記フラグを格納する第1のメモリと、上記最適化実行間隔情報を格納した第2のメモリと同一装置を有する。それぞれ上記スケジューラと、緊急最適化手段と、緊急スケジューラと、上記フ

ラグを格納する第1の緊急メモリと、緊急最適化実行間隔情報を格納する第2の緊急メモリとを有するアナログスケジューラ最適化手段を備えたものである。

【0019】第10の発明は、上記スケジューラの起動を一時停止するスケジューラ起動停止手段と、上記スケジューラ起動停止手段によるスケジューラの一時停止中に、上記アナログスケジューラ最適化手段で最適化された上記実行スケジュールを上記スケジューラが設定中の実行スケジュールと変更するスケジューラ内容変更手段とを有する最適化手段を備えたものである。

【0020】第11の発明は、上記複数の同期タスクに新たな同期タスクが追加されたときに、この新規同期タスクの固有情報を格納し、この新規同期タスクの固有情報に基づいて実行スケジュールの最適化を行い、この最適化した実行スケジュールを上記最適化手段に送るアナログスケジューラ最適化手段を備えたものである。

【0021】

【発明の実施の形態】実施の形態1. 以下、実施の形態1を例として説明する。これは、実施の形態1の複数同期実行タスクの最適化スケジュール方式を示す構成図である。

【0022】図において、1a～1cは同期タスク、2a～2cはそれぞれ同期タスク1a～1cの固有情報、3は同期タスク1a～1cの固有情報2a～2cをスケジュール制御初期に設定されたスケジュールの調整定を行つたためとした同期タスク情報テーブル、4は実際のスケジュールを行うスケジューラ、5はスケジューラがどの同期タスクを実行すべきかを指示するのに必要なスケジュール情報、6はスケジューラ中に同期タスク1a～1cの同期間隔の値を最小値にするよう、同期タスク情報を修正する最適化機構である。

【0023】ここで、固有情報2a～2cの内部構造（ここでは同期タスク1aの固有情報2a）を示す説明図である。図において、10は従来のスケジューリング機構でも存在する情報であり、11は同期タスクの名前、12は同期タスクの同期間隔、13は同期タスクの優先度、14は同期タスクの処理を実行するのに必要な同期タスク処理実行時間である。また、15は前記に追加した情報であり、16は同期間隔の欠陥を監視するの、変更しても良い値を示す監視時間、17はこの同期タスクが前回実行された時の（直前の）同期間隔である。

【0024】次に、動作について説明する。まず、基本となるスケジューラ4の動作の説明を行い、さらに本発明の動作の説明を行う。まず最初に、基本的なスケジューリングの処理として初期化処理が行われる。これは、11における、同期タスク1a～1cの固有情報2a～2cにより構成された同期タスク情報テーブル3に基いて、スケジューラ4が、スケジュール情報5の作成を行

う。この後、スケジューラ4は、自分自身が周期的に起動され、スケジュール情報5を参照し、タイムアウトした周期タスクがあればこの周期タスクの実行を行うことを繰り返す。

[0028] 図5は、スケジューラ4が行うスケジューリングの処理の流れを示すフローチャートである。まず、スケジューラ4がタイムアウトした周期タスクの有無を確認する。タイムアウトした周期タスクの確認方法としては、図1のスケジュール情報5の例に示された優先度のタイムアウトキヤムを用意し、優先度の高いキヤムから順に確認していく方法がある。ここで、図示された優先度のタイムアウトキヤムの99、100、101は優先度を表し、また、それぞれのO印は周期タスク1a～1cで行われる周期タスクを表している。

[0029] スケジューラ4がタイムアウトした周期タスクが無い場合には、処理を行わずに終了するが、スケジューラ4がタイムアウトした周期タスクが存在した場合には、スケジューラ4でこの周期タスクの実行を行う。実行処理が完了した後は、この周期タスクが同期時間間隔に再度実行されるようにスケジューリングを行う。この時、スケジューラ4は、実行が完了した周期タスクを次の同期時間間隔に実行できるように、再スケジューリングを行う。

[0030] 図6は、スケジューラ4がタイムアウトした周期タスクがあれば、これを実行し、再スケジューリングするのであるが、本実施の形態では、スケジューラ4で再スケジューリングする直前に最適化処理が必要であるかの判定を行い、スケジューラ4で最適化が必要と判定された時には、スケジューラ4で最適化の処理を行い、スケジューラ4で再スケジューリングを行う。ここで、図3の最適化で選ばれたスケジューラ4～スケジューラ106の処理は、最適化後6で行われる処理である。

[0031] 次に、最適化の処理方法について説明する。図1は、最適化後6が行う最適化の処理の流れを示すフローチャートである。このフローチャートは、図3の最適化で選ばれたスケジューラ4～スケジューラ106の処理をさらに詳しく説明したものである。最適化の処理は、まずこの周期タスクが最適化のために同期時間を超過していないかどうかを確認する。この方法としては、スケジューラ4で同期時間が0以外の数値に設定されているか調べ、スケジューラ4で同期タスク前に設定されている同期時間が0であるか0以外の数値の判断を行う。

[0032] スケジューラ4で同期時間が0の場合には、最適化の処理を行わずに、スケジューラ4で同期時間を初期値のまま決定し、処理を終える。一方、スケジューラ4で同期時間が0以外であった場合には、スケジューラ4で同期実行された時から今回の実行ま

での間隔（時間）、即ち周期タスク1毎の固有情報2にからがけで決定されている前回の同期時間と、周期タスク1毎の固有情報2にからがけで決定されている（要求されている）同期時間12との比較を行い、スケジューラ4で一意していることが確認された場合には、特に最適化を行う必要がないため、スケジューラ4で同期時間を初期値のまま決定する。

[0033] スケジューラ4で一意していないことが確認された場合には、スケジューラ4で同期時間を変更して設定を行う。変更の方法としては同期時間の値をこのまま同期時間に加算もしくは減算し、新しい同期時間間で次のスケジューリングを行う方法、または、0以上、同期時間以下の値を乱数で決め、この値を同期時間に加算し、次のスケジューリングを行う方法などが考えられる。

[0034] 最適化処理は、この再スケジューリングを行う時に実行される。一例を用いて説明すると、周期タスク情報テーブル3の中に周期タスクがタイムアウトしたかどうかを判断する必要があるタイムアウトカウンタを保持するようスケジューリングシステムでは、例えば、周期タスク1bがちょうど実行を終えたならば、周期タスク1bのタイムアウトカウンタに周期タスク1bの同期時間の値をこのままにし、指定された同期時間後に再度実行できるようにするのであるが、周期タスクが同期時間と異なると場合には、次の同期時間を変更して決定する。変更する方法としては、同期時間の最大値を加算する方法や、スケジューラ4が取り扱える最小のタイムアウトカウンタでも良い。

[0035] 当然のことながら、今回の同期時間を変更した周期タスクの場合、この次に実行された時に振り返って確認する前回の同期時間は、1周期間隔で変更となるので、一度変更を行った数に比較を行う際には、前回に行われた変更量を考慮しなければならない。例えば、前回の同期時間を保持するのに、現時点での時刻を記録し、前回の時刻との差分を求める方法が考えられるが、容易に行えるように、変更を行った時の時刻には、この変更量を加算した値を保持しておくなどが考えられる。

[0036] 以上のように、本実施の形態によれば、前回の同期時間が、所望されていた値と違った周期タスクに対して調整を行うことになり、実行間隔を最適化することができる。定期処理を実行する複数の周期タスクが実行されるときに、どの周期タスクも同期タスク前に設定された一定の間隔で定期処理が実行できるようにスケジューリングの最適化を行うことができる。

[0037] 実施の形態2。以下、実施の形態2を図を参照して説明する。本実施の形態は、図5のために変更することができる同期時間の値を複数保持し、これらの同期時間を選択使用すること、同期期に実行することが可能である実施地点（最適点）を求める処理を向上。

また、局所解（ローカルミニマ）に陥った場合に脱出可能である場合について説明する。

[0038] 図5は、固有情報2a～2cの内部構造（ここでは周期タスク1aの固有情報2a）を示す説明図であり、実施の形態1との違いは、図5における同期時間16と前回の同期時間17が複数使用されていることである。図5では、同期時間及び前回の同期時間がそれぞれ同期時間テーブル21、前同期時間テーブル22になっている。

[0039] 次に、本実施の形態では、どのようにして最適化が行われるのかについて説明する。例えば、同期時間テーブル21には同期時間を10と20の2つ（21a、21b）、前同期時間テーブル22に同期時間を10個（22a、22b、・・・）持つ同期タスクdがある。同期タスクdの同期時間を30とし、この他には同期時間が10の周期タスクa、b、そして同期時間20の周期タスクcがある。

[0040] 全ての周期タスクの優先度が同一であるとし、従来の方法でスケジューリングが行われると、図5に示すように、同期時間の低いものの順、即ち、周期タスクa、b、c、dの順にCPUが割り当てられる。すると、周期タスクa～cの同期時間は常に同一であり、維持されている値になるのであるが、一番同期時間の低かった周期タスクdは、ぶれが生じてしまう。

[0041] このような場合において、実施の形態1で説明した方法を使用した場合、同期時間の値の設定を誤ってしまえば、同期のぶれを最適化することができない場合がある。上記に説明した例で言えば、同期時間が10であるとして、いつまで経っても収束しない、そこで、同期時間を複数用意することになり、局所解からの脱出を図る。例えば、同期時間で変更しているにも関わらず、全ての前同期時間を比較すると、常に皆のぶれ幅が存在するような場合には（図5の同期時間21aの1101）を使用、最適化後6の中で、同期時間が妥当な値ではないと判断し、図5の同期時間21bの121に切り替えて処理を続ける。このように同期時間21bの121を用いて二回最適化を行うと、図7のようになり、周期タスクdの同期時間は維持されていた値になる。

[0042] 即ち、周期タスクA1aを基準にして、周期タスクA1aの1単位後に周期タスクB1bが開始される、次に周期タスクB1bの2単位後に周期タスクC1cが開始され、次に周期タスクA1aの4単位後に周期タスクB1bが開始される場合において、再度周期タスクA1aから周期タスクB1bまでの周期タスクの実行が行われる際、周期タスクA1a～周期タスクC1cの開始のタイミングが合い、同期間隔にぶれが生じることがない。

[0043] 以上のように、本実施の形態によれば、前回の同期時間の情報を複数保持し、また、同期時間を

調整することにより、最適化を行う際に局所解に陥ってしまうのを防ぎ、各周期タスクの同期時間を最適化することができるようにスケジューリングの最適化を行うことができる。即ち、定期処理を実行する複数の周期タスクが実行されるときに、どの周期タスクも同期タスク前に設定された一定の間隔で定期処理が実行できるようにスケジューリングの最適化を行うことができる。

[0044] 実施の形態3。以下、実施の形態3を図を参照して説明する。実施の形態2で、図5のために変更することができる同期時間の値を複数保持することについて説明したが、本実施の形態では、この同期時間の値を可変にすることによって、さらに最適化の改善を行うものである。

[0045] 本実施の形態の情報は、実施の形態2とはほぼ同様であり、相違点は同期時間テーブルが最大同期時間と、現在使用している同期時間の2つだけしかない。図5で説明すると、同期時間テーブル中の21aと21bのみを使用する。ここでは、21aは最大同期時間、21bは現在使用している同期時間を表している。

[0046] 処理の基本的な流れは実施の形態1と同じであり、相違点は、図5のフローチャート中のスケジューラ4で、同期時間を変更して決定（実行）を行う際に、この数値を可変にした点である。例えば、前回の同期時間と、このまた前回の同期時間とを比較し、この大小によって、次の同期時間を決定する方法である。

[0047] 図5は、同期時間を決定する処理の流れを示すフローチャートである。図5を用いてこの処理動作を説明する。まず、スケジューラ4で前回と前々回の同期時間の比較を行う。例えば、周期タスク1前に予め設定されている周期タスク1の同期時間が1501、同期時間の最大値が1101の場合には、周期タスク1が前々回の同期時間の最大値が1501と前回の同期時間が1601と前回と同期時間1601との比較を行う。

[0048] スケジューラ4で、上記の比較を行った結果、前回の前々回のより大きいため、次の同期時間121に適用する同期時間は、スケジューラ4で前回使用している同期時間の2倍になる。ただし、値は同期時間の最大値を超えないようにする。一回目で、前回の同期時間は実質101なのであるが、同期時間の最大値が使用されたため、次回も最大値が使われる。

[0049] また、二回目以降においては、前回は1601、前々回は1701、前回の同期時間は1101であった場合には、スケジューラ4で、次の同期時間は前回の半分に設定、つまり151にする。なお、二回目以降においては、前回、前々回とも1801、前回の同期時間は121であった場合には、局所解にまっ

タスクの詳細や、現状でのスケジュールリソース状況の表示
タスクの画面に出たり、このスケジュールシート情報に対し
てユーザが各時間タスクのためのスケジュールリソースのスケ
ジュールを最大可能にするためのリソース要求を強要に
表示しているような状況である、とが出力内容から読み
取れる。このようなことは、例えば、なかなか最大に
出ていることなどである、とが出力内容から読み
取れる場合、ある時間タスクの1時間単位を広くと
ることなどを試みることができる。

【0070】こうした人々のための「積極的処置」は、突然に改定スケジュール、また実行がされている間期タスクに対して行うことは、単独で構成である場合は可能であるが、多く複雑な場合は困難であるが、APR 促進強化（図 71）（ビジュアル）を介することで、より同期タスクの実行時間を変更し、ユーザが理解しやすいため（図 72）である）状況を作り出すことが可能になる。

【0071】このようにして得られた新たなパラメータ設定等の情報は、再選システムにて同種タスクを実行する際に利用することが可能である。これによって、次回実行時には最適化されたスケジューリングで実行が可能になる。

【0072】カーネル間の最適化は図6と7のアプローチ
 ョンのAP最適化は図71との間の違いには、前
 実施の形態のように命令実行手段5を介して行う。A
 P最適化は図71内には、カーネル間のスケジューラ
 4や両方スケジューラ4、スケジューラ4情報5の
 最適化ものを保持しており、カーネルで行われるス
 ケジューリングと同等の処理を行う。

[0073] 図16は、シミュレーションの処理の流れを示すフローチャートである。図において、まず、ステップS151で、現状の短期タスクの情報を取得する。獲得する方法は、命令実行手段51を介することによって、例えば、I/O命令を発生し、調査可能プログラム62及び装置に実行時間61、スケジューム情報63の格納されているメモリ領域を読み込む。

[0074] 次に、ステップS152で疑似スケジューラ72がスケジューリングを行う。疑似スケジューラ72では、カーネル間のスケジューラ4のうち、周期タスク1のスケジューリングに對する部分のみを讀した機能を持しており、上記で得た情報に基づき、どのようなスケジューリングが行われているかを計算する。

[10075] 次に、ステップS153でスキャンラインの状況をユーザが理解できるように画面側に映し出す。表示の形態は、図5や図11のようなもので、周知のスラセ及び周知部間の電位が多い場合には、最も周知部間の電位が大きいものの最大値として、表示可能な範囲で電位を表示し、後は、表示範囲の一部を遮断するなど、この部位が拡大されるようにしてあれば、ユーザ側が必要な情報だけが得られる。

[0081] 次に、ステップS162でメニューボ
ンによって得られた情報（カーネルの周波数成分6）に
て、ステップS163で再度メニューボンを押下
す。この時、例えば、メニューボンを押下し
た全情報（図4Aメニュー6が保持するメニューエ
ント情報とカーネル毎のメニュー情報8）に「なるこ
とにより」を添えることにより、即座に最速にさら
れたメニュー6が選択可能になる。また、メニューエ
ント情報は更新され、各メニューエントのメニューに
含まれる情報を更新し、最速にその内容が最適の度
数で表示することも可能である。

[illegible]

【0083】実施の形態9. 以下、実施の形態9を図を参照して説明する。本実施の形態は、7ブレード・ジョイント側にて最速化の計算を行っていき、最速解が求まってからこの結果をカーネル側の最速化回路の戻し、スラジジョイント側を行わせる際に、新たな明解スラジの追加を行った場合のシミュレーション及びこの結果をカーネル側の戻映を可能とするものである。

100841 1119は、要約の作成80%の数の短期集約プログラムの最適化システムという方式を示す特許文である。新規プログラム追加機能という1が追加されている。例えは、既存の短期プログラム1を新しいプログラムのシステムにおいて、既存の短期プログラム2を新規に追加するような場合、ユーザが操作する入力装置10を介して、APは最適化処理711の新しいプログラム追加機能911に既存プログラム12の要約のプログラムにも固有な税22を渡される。要約のプログラムは、ジョブメニューを行なう、利用し、あるいはプログラムは、ジョブメニューの結果から一方向に渡し、実行に要する際に必要になる。

【0085】本発明の形造の辺は、シェーダーの順に「新形タス」を進む部分と、シェーダー後の形造を「カネス」側に進む部分とに分かれる。図21は、新形タス追加の二つの型に分かれる。図22は、シェーダーにおいて、基本形造の流れる方向を「プロセッサ」である。図において、基本形造の流れる方向は、実際の形造8の図1における「タスS161」。「タスS151」〜「S153」の部分と一致する。唯一の違いは、タスS154で「ユーザ」側から「新形タス」の追加が指示を受けると、新形タス追加の形造（図23の「プロセス」）を「新形タス」追加の形造8 1に渡し、タスS170で図24のタスユーザ側の更新を行う点である。

【0086】新規スラグ造肉用9は従来と比べると、本装置の仕込量である時というには、入出力係数 γ が0から入出力係数 γ まで、つまり、既成スラグ造肉がスラグがスラグでない状態、新規スラグは既成スラグと既成スラグがスラグでない状態、新規スラグ造肉がスラグ2は、前記したように、カーネル側のスラグ造肉における既成スラグのサイズ γ とスラグ係数を換じたものであり、既成スラグのサイズ係数 γ を基にスラグ造肉 γ を行ってゐる。

【0087】新規タスク追加情報91は、新規の即時タスク12の固有情報22に基づき、属性スケジューリング情報76に追加を行う。例えば、図中のスケジューラ情報51に示されるように属性制御のタイムアウトキーマスケーリングが管理されている場合には新規の即時タスク12の優先度を参照し、このキーマに追加を行う。

【0088】次に、カーネル側の新情報を更新する時の動作について説明する。基本には、従来の形態と一緒であるが、新設のシステム1が追加されることで、図5のシステム1の発行を行うように要する情報を、カーネル側のシステム1が本体に伝えなければならない。これは、システム1が本体が図5のシステム1を発行するには、図5のシステム1の情報を得ない、発行プロセスの情報を得るので、今回新設追加した図5のシステム1の図5のシステム1情報（システム1の追加）しなければならないこととなるためである。

100891 この年の処理は、莫大の経費8億にて用し、
スウェーデン内政省新設第82にて、周知スラング報
デナールを用いて、内政省を組織し、この年1月頃に命
令発行する。この年にAPR内政省は第719の所収
周知スラング追加第91が保持する周知スラングのアル
ス等の清道を格納し、スウェーデン本報が参照される
ようにする法が考えられる。また、新に追加される
周知スラングを見越して、初期化処理時に周知スラング報
デナールの領域を主目に格納しておく方法も考えら
れる。

[illegible]

【00091】
【発明の効果】 この発明は、以上説明したように構成されているので、以下に示すような効果を奏する。

【0092】第1の発明では、定周期処理を実行する模

数の同期タスク毎の固有情報に基づいて、複数の同期タスクを実行する実行スケジュールの最適化を行い、この最適化された実行スケジュールに従って、予め設定された複数の同期タスクの実行スケジュールを修正することにより、定期処理を実行する複数の同期タスクを一定の同期間隔で実行できるようなスケジュールの最適化を行うことができる。

【0083】第2の項明では、複数の同期タスクを実行する第1の同期間隔情報、複数の同期タスクを実行するときに生じる第1の同期間隔情報の変化量を観測する観測間隔情報に基づいて変更する否か判定し、変更するときは第1の同期間隔と同期タスクが前回実行されたときの第2の同期間隔情報とが一致しているかを判定し、一致していないときには観測間隔情報に基づいて第1の同期間隔を変更することにより、定期処理を実行する複数の同期タスクを一定の同期間隔で実行できるようなスケジュールの最適化を行うことができる。

【0084】第3の項明では、複数の同期タスクを実行する第1の同期間隔情報、複数の同期タスクを実行するときに生じる第1の同期間隔情報の変化量を観測する複数の観測間隔情報に基づいて変更する否か判定し、変更するときは第1の同期間隔と複数の同期タスクが前回実行されたときの第2の同期間隔情報とが一致しているかを判定し、一致していないときには複数の観測間隔情報に基づいて第1の同期間隔を変更することにより、定期処理を実行する複数の同期タスクを一定の同期間隔で実行できるようなスケジュールの最適化を行うことができる。

【0085】第4の項明では、複数の同期タスクを実行する第1の同期間隔情報、複数の同期タスクが前回実行されたときの第2の同期間隔情報と複数の同期タスクが前回実行されたときの第3の同期間隔情報とに基づいて設定された観測間隔情報で第1の同期間隔を変更することにより、定期処理を実行する複数の同期タスクを一定の同期間隔で実行できるようなスケジュールの最適化を行うことができる。

【0086】第5の項明では、ニューラルネットワークにより出力された観測時間に基づいて第1の同期間隔を変更することにより、定期処理を実行する複数の同期タスクを一定の同期間隔で実行できるようなスケジュールの最適化を行うことができる。

【0087】第6の項明では、利用者により入力された命令に基づいて、上記複数の同期タスクの実行スケジュールの最適化を行うことにより、利用者からの要求で定期処理を実行する複数の同期タスクを一定の同期間隔で実行できるようなスケジュールの最適化を行うことができる。

【0088】第7の項明では、複数の同期タスク毎の同期タスクの最適化が全て終了する最適化実行時間情報に基づいて上記複数の同期タスクの実行スケジュールの最

適化を行うことにより、所定の時間内に定期処理を実行する複数の同期タスクを一定の同期間隔で実行できるようなスケジュールの最適化を行うことができる。

【0089】第8、9の項明では、利用者からの指示に基づいて複数の同期タスクの実行スケジュールを最適化することにより、利用者からの指示で定期処理を実行する複数の同期タスクを一定の同期間隔で実行できるようなスケジュールの最適化を行うことができる。

【0100】第10の項明では、スケジュールの起動を一時停止する最中に、アプリケーション側最適化手段で最適化された実行スケジュールをスケジュールが設定中の実行スケジュールと変換することにより、利用者が利用できるアプリケーション内で定期処理を実行する複数の同期タスクを一定の同期間隔で実行できるようなスケジュールの最適化を行うことができる。

【0101】第11の項明では、新規同期タスクが追加されたとき、新規同期タスクの固有情報に基づいて実行スケジュールの最適化を行い、この最適化した実行スケジュールを上記最適化手段に送ることにより、新規同期タスクが追加されたときにも複数の同期タスクを一定の同期間隔で実行できるようなスケジュールの最適化を行うことができる。

【図面の簡単な説明】

【1】 実施の形態1の複数同期実行タスクの最適化スケジュール方式を示す構成図。

【2】 実施の形態1における固有情報2a～2cの内部構造を示す説明図。

【3】 実施の形態1におけるスケジュールの処理の流れを示すフローチャート。

【4】 実施の形態1における固有情報2a～2cの内部構造を示す説明図。

【5】 実施の形態2における固有情報2a～2cの内部構造を示す説明図。

【6】 実施の形態2における固有情報2a～2cの内部構造を示す説明図。

【7】 実施の形態2における固有情報2a～2cの内部構造を示す説明図。

【8】 実施の形態3における固有情報2a～2cの内部構造を示す説明図。

【9】 実施の形態4における固有情報2a～2cの内部構造を示す説明図。

【10】 実施の形態4における固有情報2a～2cの内部構造を示す説明図。

【11】 実施の形態5の複数同期実行タスクの最適化スケジュール方式を示す構成図。

【12】 実施の形態5における最適化の処理の流れを示すフローチャート。

【13】 実施の形態6の複数同期実行タスクの最適化スケジュール方式を示す構成図。

【14】 実施の形態6における最適化の判定方法の

処理の流れを示すフローチャート。

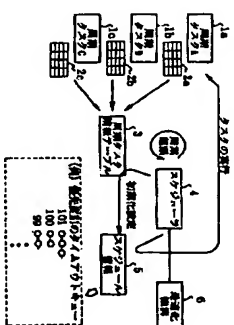
【15】 実施の形態7の複数同期実行タスクの最適化スケジュール方式を示す構成図。

【16】 実施の形態7におけるシミュレーションの処理の流れを示すフローチャート。

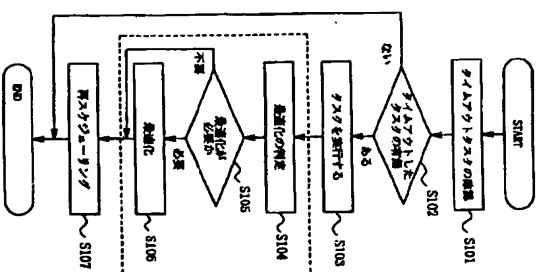
【17】 実施の形態8の複数同期実行タスクの最適化スケジュール方式を示す構成図。

【18】 実施の形態8におけるスケジュールの処理の流れを示すフローチャート。

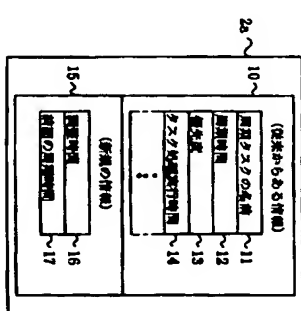
【19】



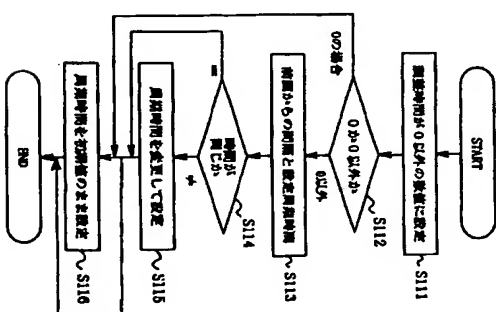
【1】



【2】



【3】



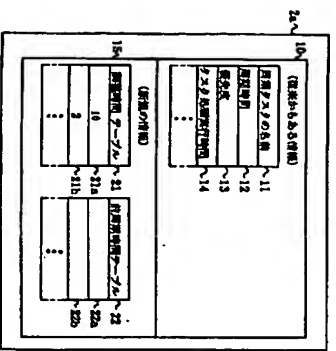
【14】 実施の形態8の複数同期実行タスクの最適化スケジュール方式を示す構成図。

【15】 実施の形態8における最適化の処理の流れを示すフローチャート。

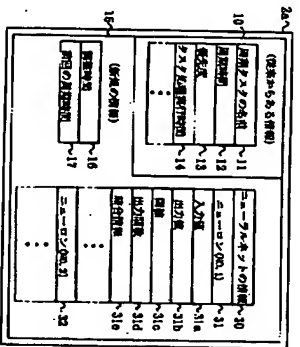
【16】 実施の形態8における最適化の判定方法の

1 固有情報、2 固有情報、3 固有情報、4 スケジュール、5 スケジュール情報、6 最適化情報。

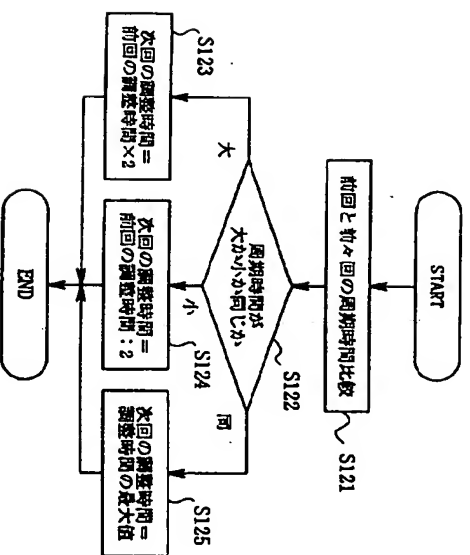
【図5】



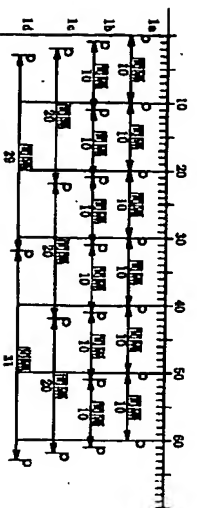
【図6】



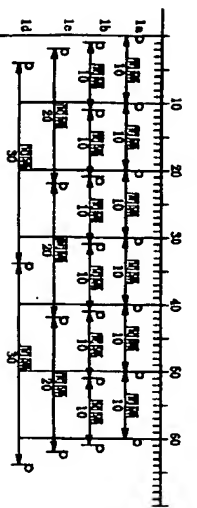
【図7】



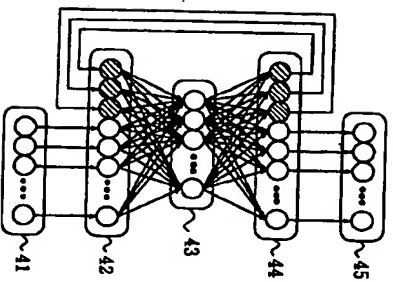
【図8】



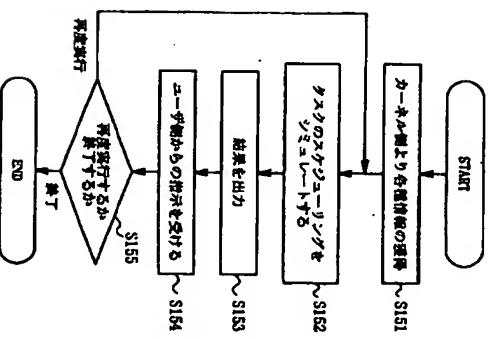
【図9】



【図10】



【図11】



THIS PAGE BLANK (USPTO)